



Mail



Reply



Reply All



Forward



Junk

Close



- Deleted Items
- Drafts
- Inbox
- Junk E-Mail
- Sent Items

[Click to view all folders](#) ▼ [Manage Folders...](#)**RE: The raw logic plugin****Xiao, He****Sent:** Thursday, February 26, 2015 5:52 PM**To:** [Rosu, Grigore](#); [Luo, Qingzhou](#); [Zhang, Yi](#)

Hi Grigore,

I also thought about inserting the raw monitoring code to the event handler, but initially, I planed to just insert the code to the precise event handlers that need to execute it with the help of additional syntax.

As far as I'm concerned, if no static variable is used, and we simply insert the whole monitoring code after "raw:" to every event handler, then the generated code may does not compile.

Think about this case:

```
SomeSpec(T t0) {
    //user can add flags manually.
    static boolean e1, e3;

    static Map<T,T1> map1;

    event e1(T t0, T1 t1) {e1=true;...}

    event e2(T t0, T2 t2) {//some operation specific
to e2}

    event e3(T t0, T1 t1) {e3=true;...}
    ...

    raw:
    if(e1 || e3) {
        map1.put(t0, t1);
    }

    ...
    //reset flags
}
```

The event action for e2 is unique, and as a result, we do not need to write its action code in "raw" block.

However, since e1 and e3 have same action, we may want to put it inside "raw" block to reduce duplication.

Here comes the problem: event e2 does not contain variable with name "t1", so it does not compile!

For the design of the "raw logic plugin":

In the monitoring code after "raw:" if we are allowed to have a global view and query/manipulate anything events in one place,

then we need to know how to refer to the arguments of each event: so there is typically more information in "raw" block.

But in a real spec, different events can have different number of parameters, with different names. Each event may only have a subset of the variables used in "raw", as a result, they won't know how to deal with the extra information.

With the help of global variable, we can solve this problem, but you have pointed out the potential drawbacks.

Without any analysis of the internal java program, I currently does not come up with a good idea about how to solve this problem.

Anyone has some idea?

Regards,
He

From: Rosu, Grigore
Sent: Thursday, February 26, 2015 2:49 PM
To: Xiao, He; Luo, Qingzhou; Zhang, Yi
Subject: RE: The raw logic plugin

He,

It is not clear from your message whether you are talking about a design, an implementation, or a use of the new raw monitor. It would definitely be a terrible idea to always keep copies of the parameters in the monitor, both from a memory efficiency point of view, and because of memory leaks (it will not be easy to garbage collect monitors). So users should keep such copies of parameters manually, if they want to and if they know what they are doing. For the beginning, forget even the flags and let people handle them manually. So all you should do is to take the Java code after "raw:" and insert it right after where you insert the event handler code, for each event.

With that in mind, your mop spec you just sent makes sense as a particular use of the intended raw monitor, although you forgot the specification name and parameters. But I hope that you did not mean to add the static variables automatically, for all raw specifications! LET THE USER DO THAT MANUALLY!

Grigore

From: Xiao, He
Sent: Thursday, February 26, 2015 12:23 PM
To: Luo, Qingzhou; Zhang, Yi
Cc: Rosu, Grigore
Subject: The raw logic plugin

Dear Qingzhou and Yi,

For the raw "logic plugin" mentioned by Grigore, I think what he wanted may be something like this:

Inside a spec, we defined static boolean variable and other variables for each event parameter.

When the event is observed, the event action method will be invoked, and we update the boolean var and

event arguments variables. At the end, we invoke the function defined after "raw:" keyword.

So an example can be:

```
static boolean e1, e2;
static T1 arg1, T2 arg2;

static Map<T1,T2> someMap;

event e1(T1 t1, T2 t2) {
    e1 = true;
    arg1 = t1;
    arg2 = t2;

    //invoke raw
}

event e2(T1 t1, T2 t2) {
    e2 = true;
    arg1 = t1;
    arg2 = t2;

    //invoke raw
}

raw:
    if (e1 || e2) {
        if (arg1.equals(SomeVal))
            someMap.put(arg1, arg2);

    //reset
    e1 = false;
    e2 = false;
}
```

The code duplication can be reduced. Previously, the event action code has to be defined in both e1 and e2.

What do yo guys think?

Thank you
He

2/26/2015

RE: The raw logic plugin - Outlook Web App, light version



Connected to Microsoft Exchange